Protocol sensing across multiple space missions

Clayton Okino, Andrew, Gray, Joshua Schoolcraft
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive M/S 238-343
Pasadena, CA 91109
{clayton.okino, andrew.gray , joshua.schoolcraft }@jpl.nasa.gov

Abstract— In this work, we present sensing performance using an architecture for a reconfigurable protocol chip for space-based applications. Toward utilizing the IP packet architecture, utilizing data link layer framing structures for multiplexed data on a channel are the targeted application considered for demonstration purposes. Specifically, we examine three common framing standards and present the sensing performance of these standards and their relative de-correlation metrics. Some analysis is performed to investigate the impact of lossy links and on the number of packets required to perform a decision with some probability. Finally, we present results on a demonstration platform that integrated reconfigurable sensing technology into the Ground Station Interface Device (GRID) for End-to-End IP demonstrations in space.

1. INTRODUCTION

In this work, we present sensing performance for a software reconfigurable network processor-based architecture for space-based communications and then present results on a demonstration of the implemented core in a test scenario. The *reconfigurable protocol chip* (RPC) is a rapid autonomous communication platform for reconfiguration of space communications network functions. Specifically, the RPC focuses on OSI model layer 2 (data link layer) detection, processing and reconfiguration. Other aspects of the RPC capture OSI model layer 1 PHY future capabilities, as well as reliable transport mechanisms (typically shared by OSI layer 4 and combined layer 1 & 2) and fault tolerance capabilities (typically required for space-based equipment). This reconfiguration capability provides for a long-life space communications infrastructure, enables dynamic operation within space networks with heterogeneous nodes, and compatibility between heterogeneous space networks (i.e. distributed spacecraft missions using different protocols) as depicted in Figure 1. This work builds upon numerous advances in commercial industry as well as NASA and military software radio developments for space network processing developments.

Dynamic reconfiguration techniques developed herein include autonomous network/protocol identification and autonomous network node reconfiguration. Both the Earth Science Enterprise Strategic Plan and Research Strategy for 2000-2010 identify satellite constellations and specifically distributed spacecraft and particularly formation flying technologies as an important technology thrust and investment areas, applicable to a range of missions. Specifically, commercial protocols might be used or might be modified for use in many future distributed spacecraft missions and various sets of missions.

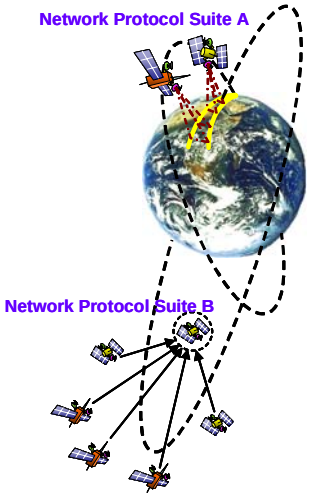

Figure 1 Heterogeneous networks in space

There are two philosophical approaches to resolving the incompatibility of protocols across a set of missions. One approach is to mandate a single link procedure for all space missions which can be problematic since missions will desire to use the approach they believe best suites their mission needs. An alternative is to enable technology that will adopt a suite of possible options and reconfigure per each of the missions link procedures. We take this later approach in this paper. Furthermore, plans for the Exploration Science Mission Directorate (ESMD) and presidential vision to return manned missions to the Moon and on to Mars will have the potential for a multitude of heterogeneous network connectivity where missions trade network and interoperability capabilities against cost constraints.

We initially present the basic concept for the architecture of the reconfigurable protocol chip. We then proceed with presenting performance results for our corresponding sensing approach.

2. Reconfigurable Protocol Chip

ARCHITECTURE

The reconfiguration architecture presented herein contains three key components required to identify and perform reconfiguration in space: 1) External stimulus detected that will either result in a requirement to perform a chip reconfiguration or a desire to reconfigure a chip; 2) Sensors used to perform the detection (possibly through in-situ processing); 3) Intelligent processing in the form of a processor that makes decisions based on output processing from the sensors and known state (condition). Figure 2 depicts the various components given the input stimulus. We now briefly describe the various components of the reconfigurable architecture where the RPC is a critical component. For further detail on the reconfigurable architecture, see [11].

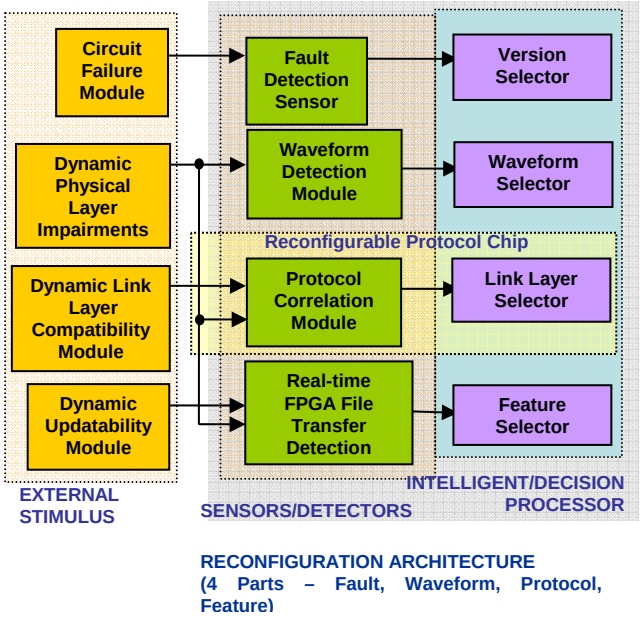

Figure 2 Space-based Reconfigurable Chip Architecture

In space-based operations, various interactions are desirable or necessary. We briefly describe a set of currently realizable or desirable sensing interactions such as radiation, physical layer communication, link layer variability (inter-heterogeneity and intra-heterogeneity), updatability (to improve overall performance or correct errors in original design).

Radiation-A key source of failure of a module in space resulting in a system fault in space environment as described in [2].

Physical Layer Communication Impairments-In space, key impairments and the effect it has on performance at the physical layer (assuming RF links) are due to variations in the channel. The effect of these impairments can be mitigated utilizing various waveforms, error correction techniques as they are implemented in a SDR platform [9].

Protocol Sensing-In terms of OSI model data link layer (layer 2), in the space community, the potential for compatibility across a number of missions is highly dependent on the mission objectives and the possibility for a number of link layer protocols is high. As a baseline capability, we assume a form of HDLC (RFC1662)[4], 802.3, or the Generic Framing Procedure (GFP) link layer framing.

Upgradability-Version upgrades, added features, and reliability of valid transfer are all desirable and in some cases required mechanisms for space-based processing.

Fault Detection Sensing-The fault detection sensor must detect and distinguish between transient and permanent faults.

3. Sensing Model and Analysis

In this section, we consider the sensing of a particular link layer framing structure and the corresponding decision circuitry.

Protocol Correlation Module-The protocol correlation module is a layer 2 sensor that is expected to detect between a set of possible protocols. The concept of heterogeneous networks in space will be driven by a number of variables outside of the scope of this work. However, we can consider target protocols that have high probability of use in future space-based networks. Among these are HDLC variants (e.g. RFC1662), 802.3, and Generic Framing Procedure (GFP). For this paper, we consider three protocols, 802.3, RFC1662 as depicted in Figure 3 and GFP as depicted in Figure 4.

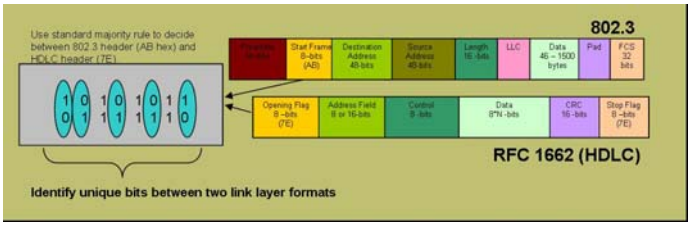

Figure 3 The 802.3 and RFC1662 (HDLC) Framing and header differences

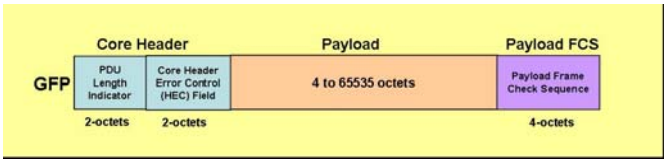

Figure 4 ITU-T G.7041/Y.1303 GFP Framing where interfaces for G.709 is specified for OTN

Link Layer Recognition and Processing Schemes for 802.3 and RFC 1662 -We assume that the physical layer is octet synchronous for both the 802.3 frame structure and the RFC1662 HDLC frame structure. Specifically, the 802.3 preamble is omitted and we focus on the 802.3 start frame delimiter and the HDLC opening flag. As in any link layer protocol some of the primary functional attributes are frame synchronization, addressing, multi-protocol selection, data transparency, and reliability. To simplify the analysis, we focus on the RFC1662. Furthermore, we assume that the address field is set at 8 bits, the control field is fixed, the Frame Check Sequence is fixed at 16-bits and we are not utilizing ARQ.

For frame synchronization, it is straightforward to perform a cross correlation between the two start field bit sequences. Recognize that 0x7E and 0xAB differ in exactly 5 bit locations as depicted in Figure 3.

Consider a generic threshold circuit that is needed to validate the start flag for a single link layer protocol. In the case of RFC1662 (or 802.3), tolerating a number of bit errors (bit flips) in the start flag would be desired. Recognize that a sensing decision circuit in the form of a threshold decision circuit used to determine if the protocol is 802.3 versus HDLC will make an incorrect decision if at least 3 of the differing bits are in error (i.e. it will mistake one protocol for the other).

Suppose p is the probability of a bit error. Then among the 5 differing bits, if any 3 or more bits are in error, then it can be shown that the sensing decision circuit will result in a protocol decision error from the binomial distribution as

$$\Pr(\text{False protocol detection}) = \sum_{i=3}^{5} \binom{5}{i} p^i (1-p)^{5-i}$$
$$= 10p^3 + 5p^4 - 14p^5$$

As depicted in Figure 5, we examine a plot for likelihood of false protocol sensing as a function of Signal to Noise (SNR) for uncoded Binary Phase Shift Keying (BPSK) modulation conditioned on reconfiguration between the two defined protocols using the simple threshold decision circuit. We observe that in general, the likelihood of a false sensing and error protocol configuration is low and decreases fast with respect to the bit error rate (BER) for BPSK. However, if the circuit is consistently monitoring on a per packet basis, and a burst of bit errors occur, then invalid reconfiguration could occur on a per packet basis. To reduce the likelihood of "protocol configuration flapping", we introduce a Markovian state based concept where we condition re-configuration on prior states.

Ideally, we would like the conditional state probability distribution of the sensing error. As an approximation, it would be advantageous to use the conditional average bit error rates.

$$\Pr(\text{error at time t} = \text{T}) = \sum_{i=3}^{5} \binom{5}{i} {p_T}^i (1-p_T)^{5-i},$$

where $p_T = p(t = T / t = 0,1,2,...,T-1)$, the average probability given the probability of the previous bit time slots. In general, one could assume that since all bits are independent, this is fixed to *p*, the probability of a bit error. However, if in a space-based (wireless) scenario the channel correlates bit errors (analogous to burst errors), then the independence assumption no longer holds and a conditional distribution is desired for state dependent autonomous protocol reconfiguration. We introduce an example of such an algorithm in [1].

We now extend the concept of error detection with higher resolution. Specifically, we consider identifying the data transparency variations within RFC1662. In particular, we detect the difference between the bit-stuff operation (RFC1662 Section 4) and the byte stuffing operation (RFC1662 Section 5). First, we briefly describe these two stuffing mechanisms and then describe a procedure for resolving the stuffing approach being used

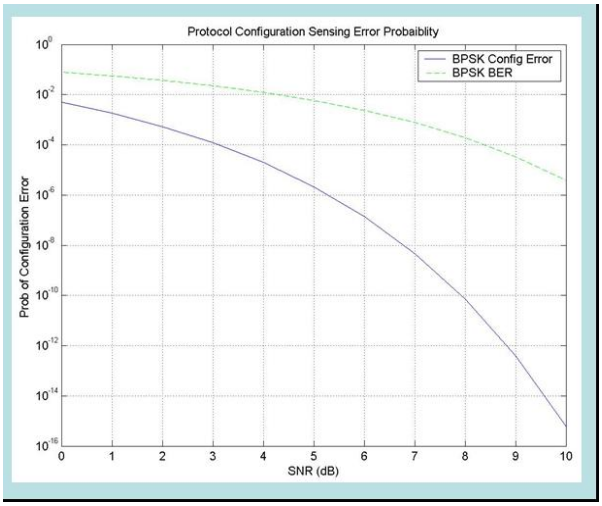

Figure 6 Protocol Sensing Error Probability

From RFC1662, for the byte-stuffing procedure, the bit sequence is examined on an octet by octet basis. Since the flag sequence is 0x7E and we assume that the likelihood is uniform among all possible octet sequences, we have the well-known result for this sequence occurring with probability 1/256. Specifically, in RFC1662 the 0x7E sequence maps to 0x7D followed by 0x5E (i.e. we have the byte sequence 0x7D5E). Another

possible character re-mapping is the control escape sequence 0x7D re-mapped to 0x7D followed by 0x5D (i.e. we have the byte sequence 0x7D5D).

From RFC1662, for the bit-stuffing procedure, the bit sequence is examined on a bit by bit basis. Since the flag sequence is 0x7E (containing five one's in a row), then a "0" bit is inserted after all five contiguous "1" bits. We have the well-known results of the likelihood of these sequences occur with probability 1/32.

In addition to utilizing the traditional CRC codes to validate that frames are correct, we can also validate using the special sequences described for the byte stuffing procedure. We assume that the only re-mapping for the byte stuffing procedure are the flag sequence and the control escape sequence. If we assume that the control escape sequence is almost never used, then we are evaluating if the bits sequence 0x7D5E exist versus the bit sequences that equate to inserting an additional "0" using bit stuffing equating to the 15-bit sequence "011111101011110". The likelihood that this is originally a bit stuffing process would be the likelihood that this exact 15-bit sequence occurred resulting is a probability of 1/2^15 = 3e-5. By executing this checking process and then weighting this scenario as a bit stuffed process with the 1/2^15 likelihood followed by the proper CRC based on detecting the end-of-frame correctly then we can select the type of stuffing. Further examination into the benefits of this procedure as oppose to simultaneously implementation of both stuffing procedures is under investigation. Note that weighting likelihood detection schemes of this form allow for a level of scalability but also present some finite likelihood of false detection.

GFP versus 802.3 and RFC 1662 (HDLC)-We now consider incorporating the GFP standard into the sensing mechanism. As depicted in Figure 4, the GFP framing procedure (as used in the ITU Recommendation for G.7041/Y.1303) involves the use of specifying a length field (PDU length field) as oppose to a start flag (used in 802.3 and RFC 1662). In addition, to strengthen the reliability of the 16-bit PDU length field, a 16-bit error checking code called the Core Header Error Control (cHEC) Field is defined. The cHEC is a CRC-16 code and defined as

$$G(x) = x^{16} + x^{12} + x^5 + 1$$

Now consider HDLC with respect to GFP. Assuming equally likely frame sizes (16-bit patterns), then there are 256 (0x7E) patterns out of the 65532 (i.e. 216 -3 -1 = 65532) possible patterns for the length field that could result in a mistaken HDLC start or rather we have 256/65532 = ~0.0039 as the likelihood (with no additional state knowledge, or use of CRC-16) of mistaking a GFP as an HDLC pattern. Similarly, we have 256 patterns out of the 65532 that can incorrectly be detected as an 802.3 start flag. To reduce this likelihood of mis-detecting the GFP framing as either an 802.3 or RFC 1662 frame, we now factor in the CRC-16 check sequence as specified in the GFP framing structure. We consider the assumption that we have a GFP frame and calculate the likelihood of mis-detecting some other start flag procedure as a GFP frame; Thus, we have 1/65532 = 1.525e-5. To reduce this likelihood of mis-detecting well below the 1e-5 region, we consider examining multiple consecutive frames. In particular, we have

$$p = \left(1.525 \times 10^{-5}\right)^n$$

where p is the likelihood of mis-detection and n is the number of consecutive frames. For n=3, the likelihood is 3.55e-15.

4. SENSING SIMULATION RESULTS

In addition to the "back of the envelope" analysis described in the previous section, simulations were performed. These include analysis of mis-detection due to other possible protocols being transmitted. Also considered was the impact of bit errors on the protocol detection process.

Matlab tests using four types of framing formats were performed on the sensing architecture model with randomly-generated frame payloads. The results, shown in Figure 7, revealed a need to establish additional differentiation with respect to bit stuffing of HDLC in the cases of GFP and byte-stuffed HDLC frames. To distinguish GFP frames, we leverage the knowledge that statistical likelihood of a mis-detect is extremely low. i.e. if we detect 2 GFP frames, the likelihood that these are random sequences and not GFP frames is on the order of 10^-10. This weighting can be factored in as a simple conditional logic statement by the intelligent decision processor.

An additional concern illustrated in Figure 7 is that the accurate determination of HDLC based on combining the probabilities of basic HDLC framing instances with stuffing instances. Specifically, in the case of transmitted HDLC byte stuffed frames, the detection mechanism from framing is currently decoupled from the stuff detection mechanism. As a follow-on enhancement, there is a need to condition the respective stuff detections to the frame detection to resolve the detection error results for HDLC byte versus bit stuffing.

Transmitted Protocol: GFP	
Sensing Protocol:	Percentage Flags Detected
GFP	**0.11%**
802.2	**0.00%**
HDLC framing ONLY	5.76%
HDLC byte-stuffed	**0.01%**
HDLC bit-stuffed	**5.41%**
Transmitted Protocol: 802.2	
Sensing Protocol:	Percentage Flags Detected
GFP	**0.03%**
802.2	**26.60%**
HDLC framing ONLY	4.42%
HDLC byte-stuffed	**0.00%**
HDLC bit-stuffed	**3.04%**
Transmitted Protocol: HDLC (bytestuffed)	
Sensing Protocol:	Percentage Flags Detected
GFP	**0.02%**
802.2	**0.00%**
HDLC framing ONLY	32.47%
HDLC byte-stuffed	**3.01%**
HDLC bit-stuffed	**18.91%**
Transmitted Protocol: HDLC (bitstuffed)	
Sensing Protocol:	Percentage Flags Detected
GFP	**0.00%**
802.2	**0.00%**
HDLC framing ONLY	2.88%
HDLC byte-stuffed	**0.00%**
HDLC bit-stuffed	**2.80%**

Figure 7 Sensing de-correlation simulation

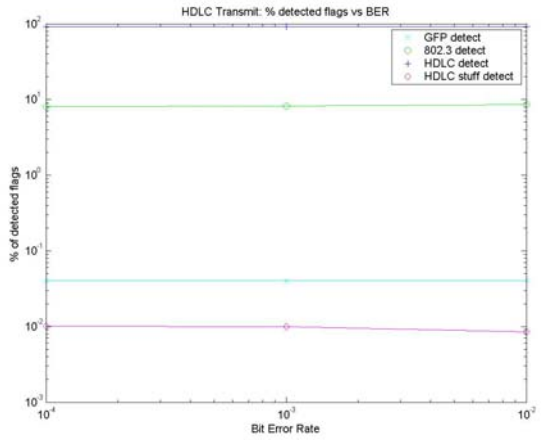

Figure 8 HDLC Sync error likelihood

Previous tests were performed to validate performance during high bit error periods. In Figures 8-10 we recognize that the sync likelihood with low false detect probability is tolerant (i.e. a flat horizontal line) of a high degree of bit errors as expected. It should be noted that the results for the GFP analysis do not capture the preamble detection capability for 802.3 and therefore have high false detection associated for the long GFP packet scenario.

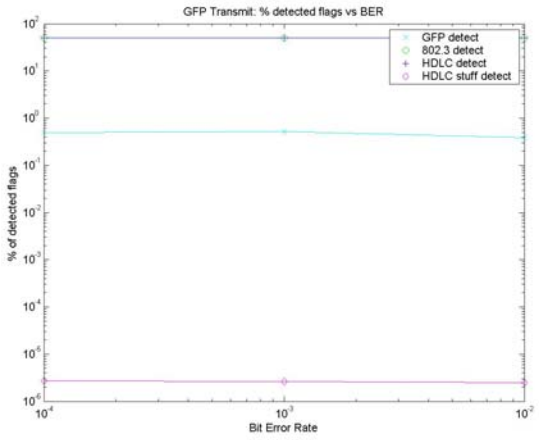

Figure 9 GFP Sync error likelihood

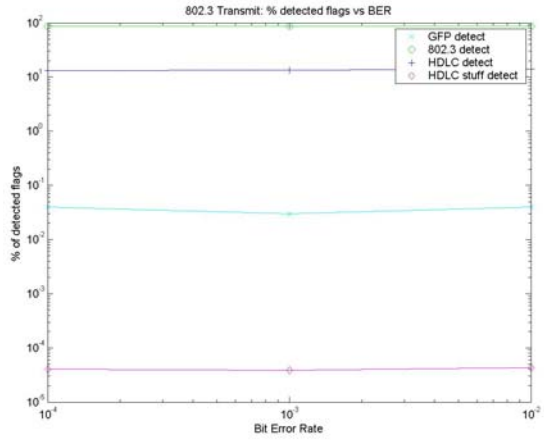

Figure 10 802.3 Sync error likelihood

We then considered the number of packets required to make a decision for each protocol. We can perform analysis on this by simply examining the relative "strengths" of each framing procedure with respect to one another. Specifically, we can obtain the probability of N fixed length packets of a protocol being mis-detected. We assume that for any given protocol that the packet sizes are of similar length and type. This is a reasonable approximation if most of the packet traffic operates based on the same types of applications that reside across each of these respective networks. We assume that bit erros are rare and so we ignore them in this analysis. Let p be the probability of a bit being transmitted. Then the probability of mis-detecting GFP in N GFP packets is (p^32)^N. Similarly, we can obtain the probability of misdetect for 802.3 as (p^16)^N and for HDLC ignoring the stuff bytes or bits) is (p^8)^N. We recognize that a per packet mis-detect likelihood for GFP is p^16N less likely than 802.3 and 802.3 is p^8N less likely of mis-

detect with respect to HDLC start frames. Thus the relative weighting for detection of each form of protocol should be on the order of these relative mis-detect ratios.

Considerable work has been performed towards multi-protocol sensing. Of significance in the work is the ability to discriminate 802.3 preambles against GFP and HDLC packets. Long GFP packets require additional knowledge that can be extracted from the relative hit percentages when a GFP packet occurs as oppose to lack of the a GFP packet where there's an order of magnitude increase in hit rate when a GFP packets is transmitted. Results also indicate a need to couple the HDLC framing to the stuffing mechanism to resolve detection between the various forms of stuffing for HDLC. Models have been developed using the Matlab SYSGEN path to cross validate simulations with development of FPGA core. The basic high level model is depicted in Figure 11 where we have implemented a sliding octet framer, the GFP detector, the HDLC detector, and the 802.3 detector with output flags indicating the hit rates.

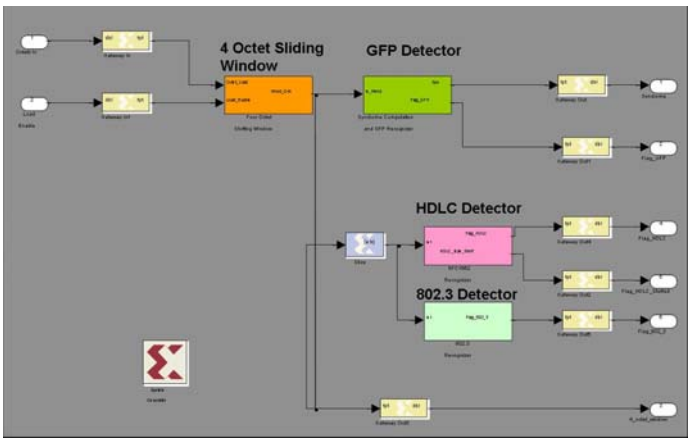

Figure 11 Protocol Sensor –Correlator

5. DEMONSTRATION

The RPC is being demonstrated on the Ground Station Interface Device (GRID) located at GSFC used for IP over space link demonstrations. Depicted in Figure 12, is the GRID card with the highlighted Xilinx Virtex 2 XC2V3000 FPGA where the RPC core resides.

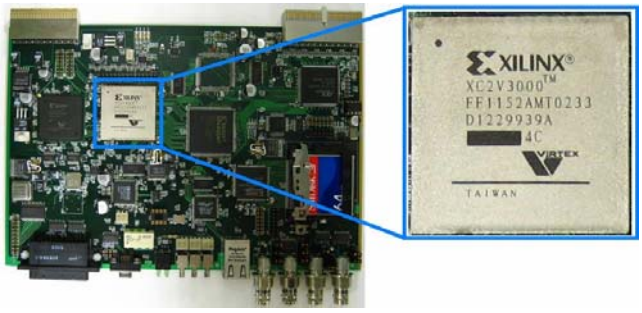

Figure 12 GRID FPGA Board

The test configuration is as depicted in Figure 13, were the RPC core is embedded within an interface wrapper and two different data sources were connected as input. Output was analyzed using a logic analyzer.

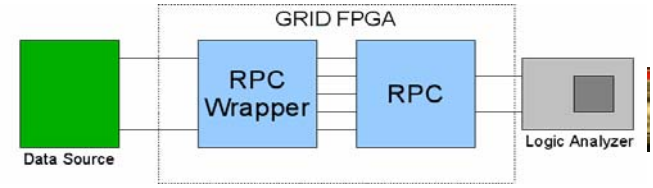

Figure 13 RPC Demonstration Test Configuration

In the first test, the RPC core was connected to a CISCO 2516 Router as depicted in Figure 14. The HDLC bitstuff idle pattern of 0x7e was recognized. The RPC "snapped" to the HDLC bitstuff stream setting when it was not set to this as the default. Sensed protocol remained locked while keep alive burst between idle patterns occurred.

- Cisco 2516 Router

Figure 14 CISCO Router connected in Test

In the second test, random bit stream patterns were generated using the GDP 615 as depicted in Figure 15. The RPC resulted in falling back into the default configuration when streams failed to match protocol correlation.

- General Data Products 615 Data Test Set

Figure 15 Random Bit Generator connected in Test

6. REMARKS

The RPC is a working product in the laboratory environment demonstrating reconfiguration of multiple protocol detection. Future work of interest is to infuse into programs such as ESMD Constellation missions. Other issues concern integrating CCSDS link layer formats. Final incorporation of this RPC core along with other reconfigurable technologies of a reconfigurable

platform is highly desirable as we transition into more dynamic networking environments.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of Chris Deleon, Dave Israel from Goddard Space Flight Center on this effort. Key to the success of the demonstration was the use of their laboratory including the platform.

CONTRACTUAL ACKNOWLEDGMENT

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology and was sponsored by the National Aeronautics and Space Administration.

REFERENCES

[1] C. Okino, C. Lee, A. Gray, P. Arabshahi, "An Autonomous Evolvable Architecture in a Reconfigurable Protocol Chip for Satellite Networks", 2003 MAPLD International conference, September 9-11, 2003, Washington, D.C.

[2] John Scarpullla and Allyson Yarbrough, "What Could go Wrong? The Effects of Ionizing Radiation on space Electronics", http://www.aero.org/publications/crosslink/summer2003/03.html

[3] Raphael Some, "Radiation Models and Hardware Design", presentation in 2002.

[4] RFC1662 – PPP in HDLC-like Framing, July 1994.

[5] Savio Chau, Adans Ko, Kar-Ming Cheung, "Mission operation for reconfigurable spacecraft", SpaceOps 2004 conference.

[6] L. Clare, J. Gao, E. Jennings, C. Okino, "Reliable Link Layer File Transfer" DRAFT technical report May 2004.

[7] Shu Lin, Philip S. Yu, "A Hybrid ARQ Scheme with Parity Retransmission for Error Control of Satellite Channels", IEEE Transactions on Communications, no. 7, July 1982 pp. 1701-1719.

[8] Roy You, "Proximity Link with Hybrid ARQ", June 2004, JPL Technical Report.

[9] Software Communications Architecture http://jtrs.army.mil/sections/technicalinformation/fset_technical.html?technical_SCA.

[10] C. Okino and Jonathan LaBroad, "A reliable data transfer architecture for a space-based protocol chip", presented at 2005 IEEE Aerospace Conference, Big Sky, MT, Mar 2005.

[11] C. Okino, Clement Lee, Andrew Gray, Payman Arabshahi, "Space-based autonomous Reconfigurable Protocol Chip", 2005 IEEE Aerospace Conference, Big Sky, MT, Mar 2005.

BIOGRAPHY

Clayton Okino received a BS in Electrical Engineering at Oregon State University in 1989, a MS in Electrical Engineering at Santa Clara University in 1993, and a Ph.D. in Electrical and Computer Engineering from the University of California, San Diego in 1998. After receiving his Ph.D., Dr. Okino accepted a position as an assistant professor in Thayer School of Engineering at Dartmouth College, where he pursued research in communication and wireless networks, emphasizing on performance and security. In 2001, Dr. Okino accepted a position as a Senior Member of the Technical Staff in the Digital Signal Processing group at Jet Propulsion Laboratory and is now in the communications network group, where his current research is in wireless network routing and access algorithms, reconfigurable sensors, wireless QoS and location based processing techniques and has PI-ed numerous projects.

Andrew Gray has over ten years of experience in technology development in the areas of RF and optical communications and radar systems and is currently the group supervisor of the Advanced Signal Processing Projects Group at the Jet Propulsion Laboratory (JPL). Most recently Dr. Gray has been the cognizant engineer for the multi-channel broadband optical receiver for the Mars Laser Communications Project (MLCD). From June 1998 until March 2004 he held the position of senior member of technical staff at JPL. Dr. Gray completed a MBA and a PhD at the University of Southern California in May 2004 and May 2000, respectively. He completed the MS in electrical engineering from Johns Hopkins University in May of 1997 and the BS in electronics with a minor in Mathematics from Pittsburg State University in May of 1994.

Joshua Schoolcraft received a B.S. in computer and electrical engineering with a minor in computer science from the University of Maine in 2005. During his undergraduate career, Josh worked at the Jet Propulsion Laboratory via the Maine Space Grant Consortium, and is currently a member of lab technical staff. Research interests include interplanetary networking protocol development and related hardware concepts.